

Nonlinear differential algorithm to compute all the zeros of a generic polynomial

Francesco Calogero

Physics Department, University of Rome "La Sapienza", 00185 Rome, Italy
Istituto Nazionale di Fisica Nucleare, Sezione di Roma

Abstract

A simple algorithm to compute all the zeros of a generic polynomial is proposed.

1 Introduction

Notation. Hereafter, for definiteness, we always refer to *monic* polynomials of arbitrary order N ($N \geq 2$),

$$P_N(z; \vec{c}, \underline{x}) = z^N + \sum_{m=1}^N (c_m z^{N-m}) = \prod_{n=1}^N (z - x_n) ; \quad (1)$$

the *complex* variable z is the argument of the polynomial, indices such as n, m, ℓ run from 1 to N (unless otherwise indicated, see below), the N -vector \vec{c} has the N *coefficients* c_m of the polynomial (1) as its N components, \underline{x} is the *unordered* set of the N *zeros* x_n of the polynomial (1), and we assume all variables to be *complex* (unless otherwise explicitly indicated, see below). We call *generic* any polynomial the *coefficients* and *zeros* of which are *generic complex* numbers, and in particular feature *zeros* which are *all different among themselves*, $x_n \neq x_m$ if $n \neq m$. Note that the notation $P_N(z; \vec{c}, \underline{x})$ is somewhat redundant, since this monic polynomial can be identified by assigning *either* its N coefficients *or* its N zeros; indeed the N coefficients c_m can be expressed in terms of the N zeros x_n via the following standard formula

$$c_m = (-1)^m \sum_{n_1 > n_2 > \dots > n_m = 1}^N (x_{n_1} x_{n_2} \dots x_{n_m}) , \quad (2a)$$

so that

$$c_1 = -(x_1 + x_2 + \dots + x_N) , \quad (2b)$$

$$\begin{aligned} c_2 = & (x_1 x_2 + x_1 x_3 + \dots + x_1 x_N) \\ & + (x_2 x_3 + x_2 x_4 + \dots + x_2 x_N) + \dots \\ & + (x_{N-2} x_{N-1} + x_{N-2} x_N) + x_{N-1} x_N , \end{aligned} \quad (2c)$$

and so on. On the other hand, while the assignment of the N *coefficients* c_m determines the N *zeros* x_n —uniquely, up to permutations—of course *explicit* formulas to accomplish generally this task *only* exist for $N \leq 4$. ■

The investigation of the properties—and of techniques for the numerical computation—of the N zeros x_n of a polynomial of degree N defined via the assignment of its N coefficients c_m (see (1)) is a problem that has engaged mathematicians since time immemorial. In this paper a simple nonlinear differential algorithm suitable to compute numerically *all* the N zeros of a *generic* polynomial of arbitrary degree N is described; I was unable to find a previous description of this algorithm in the literature, but I am aware that my search has not been—indeed, it could not have been—quite complete. This algorithm is described in the following Section 2 and proven in Section 3.

2 Results

It is now convenient to introduce an additional independent variable t , which is hereafter assumed to be *real* and might be interpreted as *time*. Hence the above notation is now extended by writing, in addition to (1), the analogous formula

$$p_N(z; \vec{\gamma}(t), \underline{y}(t)) = z^N + \sum_{m=1}^N [\gamma_m(t) z^{N-m}] = \prod_{n=1}^N [z - y_n(t)] , \quad (3)$$

to which notational comments quite analogous to those reported above apply.

There holds then the following

Proposition. Consider the following system of N nonlinear first-order differential equations satisfied by the N zeros $y_n(t)$ of the polynomial (3):

$$\dot{y}_n(t) = - \left\{ \prod_{\ell=1, \ell \neq n}^N [y_n(t) - y_\ell(t)]^{-1} \right\} \sum_{m=1}^N \left\{ [c_m - \gamma_m(0)] [y_n(t)]^{N-m} \right\} . \quad (4a)$$

Here and below a superimposed dot denote a t -differentiation, while the *coefficients* c_m are those of the polynomial $P_N(z; \vec{c}, \underline{x})$, see (1), the *zeros* x_n of which we seek, and $\gamma_m(0)$ are the N *coefficients* of the polynomial $p_N(z; \vec{\gamma}(t), \underline{y}(t))$, see (3), at $t = 0$, hence they are related to the "initial" values $y_n(0)$ of the *zeros* of this polynomial by the formula (analogous to (2))

$$\gamma_m(0) = (-1)^m \sum_{n_1 > n_2 > \dots > n_m = 1}^N [y_{n_1}(0) y_{n_2}(0) \cdots y_{n_m}(0)] . \quad (4b)$$

Then

$$x_n = y_n(1) . \quad \blacksquare \quad (4c)$$

It is thus seen that the *zeros* x_n of the polynomial $P_N(z; \vec{c}, \underline{x})$, see (1), can be computed—once the N coefficients c_m of this polynomial have been assigned—via the following procedure. *Step one:* choose (*arbitrarily!*) N *complex* numbers $y_n(0)$. *Step two:* compute, via the formulas (4b), the N quantities $\gamma_m(0)$. *Step three:* integrate (numerically) the system of differential equations (4a) from

$t = 0$ to $t = 1$, starting from the N initial data $y_n(0)$, getting thereby the N values $y_n(1)$, which give the sought result, see (4c).

Will this procedure always work? The only possible snag is that the solution $\vec{y}(t)$ of the "dynamical system" (4a) run into a singularity during its evolution from $t = 0$ to $t = 1$. The only mechanism whereby this might occur is because during this evolution two different coordinates $y_n(t)$ might coincide, $y_\ell(t) = y_n(t)$ for $\ell \neq n$, at some value of the *real* variable t in the interval $0 < t < 1$, causing the right-hand side of (4a) to blow up. This "collision" might indeed happen, but it is *not* a *generic* phenomenon: hence it will be enough to change the assignment of the (arbitrary!) initial data $y_n(0)$ to avoid this difficulty; note however that this suggests that to apply this method it will be advisable to always start with *complex* initial data $y_n(0)$, even in the case of *real* polynomials with *real* zeros. And note that by performing the numerical integration of the differential equations (4a) with different initial data $\vec{y}(0)$ provides moreover a possibility to assess the *numerical accuracy* of the computation, by comparing the results obtained starting from different sets of initial data.

Remark. It is plain that this procedure will work more efficiently the closer the, arbitrarily chosen, initial values $y_n(0)$ are to the N zeros x_n the values of which one is trying to compute; indeed if the N initial values $y_n(0)$ happened to *coincide* with the N zeros x_n , $y_n(0) = x_n$, this would imply $\gamma_m(0) = c_m$ (compare (2) with (4b)) hence the right-hand side of the differential equations (4a) would vanish identically, entailing $\dot{y}_n = 0$ hence $y_n(1) = y_n(0) = x_n$, consistently with (4c).

Let us also emphasize that the dependence (via (4b)) of the right-hand sides of the differential equations (4a) upon the initial values $y_n(0)$ of the dependent variables $y_n(t)$ implies that these differential equations are rather Differential Functional Equations than Ordinary Differential Equations; but this fact has hardly any relevance on *step three* of the procedure, see above. ■

A comparison of the actual effectiveness of this technique with that of other methods to compute *all* the N zeros of a *generic* polynomial of arbitrary degree N is beyond the scope of this short communication, and in any case it is a task to be rather pursued by specialists in numerical analysis if they consider it worthy of their attention.

3 Proof

The proof of the above **Proposition** is actually quite easy (raising thereby some doubts on the novelty of this finding). The starting point is the *identity*

$$\dot{y}_n(t) = - \left\{ \prod_{\ell=1, \ell \neq n}^N [y_n(t) - y_\ell(t)]^{-1} \right\} \sum_{m=1}^N \left\{ \dot{\gamma}_m(t) [y_n(t)]^{N-m} \right\}, \quad (5)$$

valid for any t -dependent polynomial with *zeros* $y_n(t)$ and *coefficients* $\gamma_m(t)$,

see (3); for a proof of this formula see [1]. Now make the assignment

$$\gamma_m(t) = \gamma_m(0) + [c_m - \gamma_m(0)] t , \quad (6a)$$

consistent with the initial (arbitrary) assignment at $t = 0$ and clearly implying

$$\dot{\gamma}_m(t) = c_m - \gamma_m(0) , \quad (6b)$$

$$\gamma_m(1) = c_m . \quad (6c)$$

The insertion of the first of these two formulas, (6b), in (5) yields (4a); while the second, (6c), implies that, at $t = 1$, the polynomial $p_N(z; \vec{\gamma}(t), \underline{y}(t))$, see (3), coincides with the polynomial $P_N(z; \vec{c}, \underline{x})$, see (1), hence the validity of (4c). Q. E. D.

References

- [1] F. Calogero, “New solvable variants of the goldfish many-body problem”, Studies Appl. Math. (in press, published online 07.10.2015). DOI: 10.1111/sapm.12096.